

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/380854967>

FRUIT SHOP MANAGEMENT SYSTEM PROJECT REPORT.

Research Proposal · August 2023

DOI: 10.13140/RG.2.2.36431.09127

CITATIONS

0

READS

373

1 author:



Kamal Acharya

Tribhuvan University

192 PUBLICATIONS 2,699 CITATIONS

SEE PROFILE

**AN
INTERNSHIP REPORT
ON
FRUIT SHOP MANAGEMENT SYSTEM PROJECT
BY
KAMAL ACHARYA
(Tribhuvan University)**

Date: 2023/08/10

1. INTRODUCTION

1.1 ABOUT THE PROJECT

The export maintenance system is a fully featured application that can help we manage fruit delivery business and achieve more control and information at a very low cost of total ownership.

A fruit export maintains automatically monitors purchase, sales, supplier information. The system includes receiving fruit from the different supplier. Customer order is placed in the system, based on the order fruit has been sales to the customer.

The report contains the details about product, purchase, sales, stock, and invoice. The main objective of this project is to computerize the company activities and to provide details about the production process at the fruit export maintenance system.

The demand of fresh fruit fruits and processed food items in international and domestic market has shown a decent increase. This estimation is creating a necessity for growing more and more fruit fruits to cater the growing demand of domestic & international market.

The customers effectively and hence help for establishing good relation between customer and fruit shop organization. It contains various customized modules for effectively maintaining fruit and stock information accurately and safely.

When the fruits are sold to the customer, stock will be reduced automatically. When a new purchase is made, stock will be increased automatically. While selecting fruits for sale, the proposed software will automatically check for total number of available stock of that particular item, if the total stock of that particular item is less than 5, software will notify the user to purchase the particular item.

The proposed project is developed to manage the fruit shop in the fruits for shop. The first module is the login. The admin should login to the project for usage. The username and password are verified and if it is correct, next form opens. If the username and password are not correct, it shows the error message.

1.2 SYSTEM SPECIFICATION

1.2.1 HARDWARE CONFIGURATION

- System : HCL/HP any
- Processor : Intel core 2 Duo
- RAM : 2 GB
- Internal memory capacity : 500 GB

1.2.2 SOFTWARE CONFIGURATION

- Operating system : Windows 7/10
- Front end : Visual studio 2008
- Back end : SQL server 2005

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The model or methodology that is still being used are defined the existing model. The existing models used in fruit shop are just the pen and paper work, so every paper should be filed thus this leads to a hectic problem when the report is needed to be generated, thus when the man power increases there is chances of getting lots of errors. So it is an intelligent plan to get upgraded to a management system. But whereas the large scale fruit shop are practical using software but these software are limited in options.

2.1.1 DRAWBACKS

- Lots of man power.
- Importance is given only for billing, where as the focus is less importance on management side.
- Orders are managed via phone calls only.
- Though only stock is focused here, there is no special alert system for this.
- Less security easily modified or delete the price list.

2.2 PROPOSED SYSTEM

The proposed system is maintains fruit shop let you know when it is time to stock-up an items that are running low and can also prevent you from the shop how the ability to keep on accurate record of the number of supplier in the stock room. A complete listing of the stock of raw materials, fruit shop contains details or records of the products quantities and stock location of companies assets. Also, it is convenient for the management to check the quality of the product. It also controls the decision of the seller whether the stocks are still sufficient for the customers.

2.2.1 ADVANTAGES

- Income and Expenditure Maintenance.
- The Stock information per day and per month can be known.
- Authenticate people only access the software.
- Price and stock update immediately & user friendly software.

SOFTWARE FEATURES

VISUAL BASIC.NET

Visual Basic.Net has revolutionized windows programming windows programming and with an object based, event driven approach to software designs. Visual basic.Net applications act as a front end to the database. Visual basic.Net application provides the interface between the user and the database. Sophisticated features that make the language truly object oriented and interfaces it with the latest in the database technology.

.NET provides a new, object-oriented API as a set of classes that will be accessible from any programming language. This book describes this framework of classes and provides a reference to what is available and how you can use this framework to write Windows applications in the brave new world of .NET.

Microsoft .NET Framework is a computing platform for developing distributed applications for the Internet. Following are the design goals of Microsoft .NET Framework:

- To provide a very high degree of language interoperability
- To provide a runtime environment that completely manages code execution
- To provide a very simple software deployment and versioning model
- To provide high-level code security through code access security and strong type checking
- To provide a consistent object-oriented programming model
- To facilitate application communication by using industry standards such as SOAP and XML.
- To simplify Web application development

Visual basic.net lets the user to add menus, text boxes, command buttons, option buttons, check boxes, list boxes, scroll bars, and file directory boxes to blank windows. Visual basic.net has many different tools.

The Common Language Runtime

The CLR is the mechanism through which .NET code is executed. It is built upon a single, common language—IL—into which source languages are compiled and includes mechanisms for executing the compiled code. This includes code verification and just-in-time (JIT) compilation, garbage collection and enforcement of security policies, and the provision of profiling and debugging services.

The CLR provides a lot of added value to the programs it supports. Because it controls how a .NET program executes and sits between the program and the operating system, it can implement security, versioning support, automatic memory management through garbage collection, and provide transparent access to system services.

The user interface is the part of the program that responds to the key press and mouse clicks. The action is referred to as events of the form and controls in the form. The form is a visual object where a set of object controls is placed to communicate with backend databases and validation checks. VB.NET provides a vast properties and methods for each controls, which helps to utilize all them, functions for record manipulations.

IMPORTANT FEATURES:

- The application is a graphical user interface.
- Client-Server architecture benefits picture and image box can be easily handled using bit mapped files and icons.
- Bit mapped files and icons are used as simple debugging tools.
- With the advent of .NET, Microsoft has introduced many new technologies that make writing component-based distributed systems easier, more flexible, and more powerful than ever before.
- It is now easier than it has ever been to write components in any programming language that can interoperate with components on other machines, which may not be Windows-based at all.

SQL SERVER 2005

The purpose of this document is to help you migrate your applications when you are migrating the underlying database from Microsoft SQL Server to Some other Database. Most of the issues encountered when migrating applications to use an Sql Server Database are related to database incompatibility. This paper presents these incompatibilities and provides solutions for many issues. It familiarize yourself with this document prior to migrating your databases. The choices made about how to migrate your application affect how you migrate the underlying database from Microsoft SQL Server to some other Database.

If the application uses the Microsoft OLE DB Provider for SQL Server or a third-party OLE DB Provider to connect the Microsoft SQL Server database, use the Sql Server Provider for OLE DB (SQLOLEDB) to connect to the migrated Database.

There are quite a few reasons, the first being that SQL is a feature rich program that can handle any database related task you have. You can create places to shop your data build tools that make it easy to read and modify your database contents, and ask questions of your data. SQL is a relational database, a database that shops information about related objects. In MS SQL that database means a collection of tables that hold data. It collectively shops all the other related objects such as queries, forms and reports that are used to implement function effectively.

SQL Server Security

- Login Authentication.
- Windows NT Authentication
- SQL Server Authentication
- Permissions validation on user database.
- T-SQL statements sent to SQL Server.
- SQL server checks user permissions on receipt of T-SQL statements

FEATURES

- Created by Microsoft and Sybase in the 80s.
- Is SQL Compliant - Uses ANSI SQL
- Supports SQL – 92 standards - Uses T-SQL
- Shops data in a central location and delivers it to clients on request
- New Server Architecture
- Graphic Administration Tools
- Maintains ANSI standards and 6.x Compatibility
- Data integrity means reliability and accuracy of data.
- Integrity rules keep data consistent.
- Supports Client/Server model.
- Request response dialog.
- Workload is split between the client and the server.
- Operating System compatibility.
- Runs on Win 95/98 NT, Netware, UNIX, OS/2, AppleTalk, Banyan VINES.
- SQL Server must have Service Pack 4 (SP4) to run on Windows NT 4.0.
- Multiple protocol compatibility.

3. SYSTEM DESIGN AND DEVELOPMENT

System design is the process of planning a new system to complement or altogether replace the old system. The purpose of the design phase is the first step in moving from the problem domain to the solution domain. The design of the system is the critical aspect that affects the quality of the software. System design is also called top-level design. The design phase translates the logical aspects of the system into physical aspects of the system.

3.1 TABLE DESIGN

Database design is a collection of interactive data shop. It is an effective method of defining, storing and retrieving the information in the database. The database design is independent of any relational database management system and it is a logical model. The logical design is mapped according to RDBMS used for implementation. The data contained in the database can be multiple application and users. It prevents the unauthorized from accessing data and ensures the privacy of data.

The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Not all of these steps will be necessary in all cases. Usually, the designer must:

- Determine the data to be shopd in the database
- Determine the relationships between the different data elements
- Superimpose a logical structure upon the data on the basis of these relationships.

3.2 INPUT DESIGN

Input design is the process of converting the user-oriented. Input to a computer based format. The goal of the input design is to make the data entry easier , logical and free error. Errors in the input data are controlled by the input design. The quality of the input determines the quality of the system output.

The entire data entry screen is interactive in nature, so that the user can directly enter into data according to the prompted messages. The users are also can directly enter into data according to the prompted messages. The users are also provided with option of selecting an appropriate input from a list of values. This will reduce the number of error, which are otherwise likely to arise if they were to be entered by the user itself.

Input design is one of the most important phase of the system design. Input design is the process where the input received in the system are planned and designed, so as to get necessary information from the user, eliminating the information that is not required. The aim of the input design is to ensure the maximum possible levels of accuracy and also ensures that the input is accessible that understood by the user.

The input design is the part of overall system design, which requires very careful attention. If the data going into the system is incorrect then the processing and output will magnify the errors.

The objectives considered during input design are:

- Nature of input processing.
- Flexibility and thoroughness of validation rules.
- Handling of properties within the input documents.
- Screen design to ensure accuracy and efficiency of the input relationship with files.
- Careful design of the input also involves attention to error handling, controls, batching and validation procedures.

Input design features can ensure the reliability of the system and produce result from accurate data or they can result in the production of erroneous information.

3.3 OUTPUT DESIGN

Output design is very important concept in the computerized system, without reliable output the user may feel the entire system is unnecessary and avoids using it. The proper output design is important in any system and facilitates effective decision-making. The output design of this system includes various reports.

Computer output is the most important and direct source of information the user. Efficient, intelligible output design should improve the system's relationships with the user and help in decision making. A major form of output is the hardcopy from the printer.

Output requirements are designed during system analysis. A good starting point for the output design is the data flow diagram. Human factors reduce issues for design involved addressing internal controls to ensure readability.

3.4. DATABASE DESIGN

A database should provide integration, Integrity and a data independence table in a database contains information pertaining to a specific entity. To maintain the tables in an effective way, it should be normalized to ensure that the number of tables does not exceed the optimum level unless it is mandatory.

To prevent unauthorized access, security measures have been provided. This may prevent unauthorized persons using data that is private. The normalization techniques have been used to design the table such that the use of all the tables is made easy.

The various relations between different tables, the number of fields in each table and the type, width of each field were analysed. The names of the fields and tables were so chosen that the users would not face any problem in identifying the table structure.

The various relations between different tables, the number of fields in each table and the type, width of each field were analysed. The names of the fields and tables were so chosen that the users would not face any problem in identifying the table structure.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective of database design is to make the data access easy, inexpensive and flexible to the user.

3.5 CODE DESIGN

Code is an ordered collection of symbols designed to provide unique identification of an entry or attribute. Sometimes used in the place of name of the item they can be specified all object's physical or performance characteristics or operational instructions. They can also show interrelationship and may sometime be used to achieve secrecy or confidentiality.

Most computer systems are stable from the compiler down to the execution of binary instructions. Therefore, it's natural to think of "product" as the artifact just above that base. That's language source code. The system that produces that artifact is still quite unpredictable, so it's not likely we'll shift our orientation. Artifacts more abstract, be they whatever, will constitute a nebulous range of "designs" or "specifications". I think it's the lack of a direct and strictly repeatable translation of these artifacts that characterizes them.

3.6 SYSTEM DEVELOPMENT

3.6.1 DESCRIPTION OF MODULES

Customer Details

This module is used for maintain the customer details like customer id, customer name, contact no, address, mail id, etc.

Fruit Details

This module is used to maintain the all fruit varieties in Fruit shop. This module which includes fruit code, fruit name, rate etc.

Supplier Details

In this option we can add, edit or delete a specified Supplier details. The add new option is used to add new details of supplier. The supplier details like supplier id, supplier name, supplier address details is to be changed then edit for modification.

Purchase Details

This is purchase module from where all procedures take place. Only through this module purchase like purchase id, supplier id, supplier name, item name, quantity, rate etc.

Sales Details

This module is used to maintain the sales details like sale id, fruit id, fruit name, customer id, customer name, quantity, rate etc.

Stock Details

This module is maintained for stock details .Stock levels are updated by purchase and sale of the product.

Billing Details

Billing process is included in sales and it used to invoice and only the registered customer can invoice and also through this module user for participation that is going to take place shortly.

4. SYSTEM TESTING AND IMPLEMENTATION

4.1 SYSTEM TESTING

Software testing is a critical element of software quality assurance and represents the ultimate reviews of specification, design and coding. Testing presents an interesting anomaly of the software. During earlier definition and development phases, it was attempted to build software from abstract concept to a tangible implementation.

The testing phase involves the testing of the developed system using various set data. Presentation of test data plays a vital role in system testing. After preparing the test data the system under study was tested using test data. While testing the system by using test data errors were found and corrected. A series of tests were performed for the proposed system before the system was ready for implementation. The various types of testing done on the system are:

Testing methodology

- Unit Testing
- Integration Testing
- Validation Testing
- White box testing
- Black box testing

4.1.1 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software design, the module. It comprises the set of test performed by the programmer prior to integration of the unit into larger system. The testing was carried out during the coding stage itself. In this step each module is found to be working satisfactorily as regards to the expected output from the module.

4.1.2 INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover error associated within the interface. The objective is to take unit tested modules and build a program structure that has been dictated by design. All modules are combined in this step. The entire program is tested as whole. And chaos in interfaces may usually result. A set of errors is encountered in such a case.

4.1.3 VALIDATION TESTING

Here in the validation testing we want to check whether the given conditions to the text box are working correctly. Because in the name place we want to enter the characters and the special symbols only we should not enter the numbers in the name field. Here while on runtime we entered numeric values in the string specified columns of product inwards. It raises error. In this phase each module has been tested by wrong inputs, for example Employee Name should be a character as well as their age should be in numbers.

4.1.4 WHITEBOX TESTING

White box testing, sometimes called glass-box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases.

- Guarantee that all independent paths within a module have exercised at least once.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to ensure their validity.

4.1.5 BLACK BOX TESTING

Black box testing, also called behavioral testing, focuses on the functional requirements of the software. That is, black box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black box testing is not an alternative to white box techniques. Rather it is a complementary approach that is likely to uncover a different class of errors than white box methods. Black box testing attempts to find errors in the following categories.

After preparing the test data the system under study was tested using test data. While testing the system by using test data errors were found and corrected. A series of tests were performed for the proposed system before the system was ready for implementation.

4.2 SYSTEM IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into a working system. The most crucial stage is achieving a successful new system and giving a user confidence in that the new system will work efficiently and effectively in the implementation stage. The stage consist of

- Testing a developed program with sample data
- Detection and correction of error
- Making necessary changes as desired by users.
- Training user personal

Visual Basic .NET (VB.NET) is a multi-paradigm, object-oriented programming language, implemented on the .NET Framework. Microsoft launched VB.NET in 2002 as the successor to its original Visual Basic language. Although the ".NET" portion of the name was dropped in 2005, this article uses "Visual Basic [.NET]" to refer to all Visual Basic languages releases since 2002, in order to distinguish between them and the classic Visual Basic, it is one of the two main languages targeting the .NET framework.

4.3 SYSTEM MAINTENANCE

Maintenance is actually implementation of the review plan as important as it is programmers and analyst is to perform or identify with him or herself with the maintenance. There are psychologically personality and professional reasons for this. Analyst and programmers spend fair more time maintaining programmer then they do writing them Maintenances account for 50-80% of total system development. Maintenance is expensive.

.NET Framework began as proprietary software, although the firm worked to standardize the software stack almost immediately, even before its first release. Despite the standardization efforts, developers, mainly those in the free and open-source software communities, expressed their unease with the selected terms and the prospects of any free and open-source implementation, especially regarding software patents. Since then, Microsoft has changed .NET development to more closely follow a contemporary model of a community-developed software project, including issuing an update to its patent promising to address the concerns.

5. CONCLUSION AND FUTURE ENHANCEMENT

5.1 CONCLUSION

This system, being the first we have created in VB.NET, has proven more difficult than originally imagined. While it may sound simple to fill out a few forms and process the information, much more is involved in the selection of applicants than this. Every time progress was made and features were added, ideas for additional features or methods to improve the usability of the system made themselves apparent. Furthermore, adding one feature meant that another required feature was now possible, and balancing completing these required features with the ideas for improvement as well as remembering everything that had to be done was a project in itself.

5.2 FUTURE ENHANCEMENT

As a future enhancement this project is about to become a web application with complete features. According to developer it can be developed as mobile application.

BIBLIOGRAPHY

REFERENCE BOOKS

1. Elias Awath, "SYSTEM ANALYSIS AND DESIGN", Tata Mc Graw Hill Publication, Sixth Edition, 2003
2. S. Ramachandran, "COMPUTER AIDED DESIGN", Air Walk Publication, Third Edition, 2003
3. Richard Fairley, "SOFTWARE ENGINEERING CONCEPTS", Tata Mc Graw Hill Publication, Second Edition, 1997
4. Programming VB.NET: A Guide for Experienced Programmers by Gary Cornell, Jonathan Morrison
5. Learning VB.NET Through Applications by Clayton Crooks II
6. VB .NET How to Program (2nd Edition) by Harvey M. Deitel, Paul J. Deitel, Tem R. Nieto.

ONLINE REFERENCES

1. www.msdn.microsoft.com
2. www.microsoftdotnet.net
3. www.dotnet-tricks.com
4. www.sqlmag.com

RESEARCHER REFERENCES

- Acharya, Kamal. "STUDENT INFORMATION MANAGEMENT SYSTEM." *Authorea Preprints* (2023).
- Acharya, Kamal. "Library Management System." Available at SSRN4807104 (2019).
- ACHARYA, KAMAL, et al. "LIBRARY MANAGEMENT SYSTEM." (2019).
- Acharya, Kamal. "Online bus reservation system project report." *Authorea Preprints* (2024).
- Acharya, Kamal. "Online bus reservation system project report." (2024).
- Acharya, Kamal. "Online Bus Reservation System." *SSRN ElectroNIC ASIA Journal* (2024): n. pag.
- Acharya, Kamal. "Student Information Management System Project." *SSRN ElectroNIC ASIA Journal* (2024): n. pag.
- Acharya, Kamal. "ATTENDANCE MANAGEMENT SYSTEM." *International Research Journal of Modernization in Engineering Technology and Science* (2023): n. pag.
- Acharya, Kamal. "College Information Management System." *SSRN ElectroNIC ASIA Journal* (2024): n. pag.
- Acharya, Kamal, Attendance Management System Project (April 28, 2024). Available at
SSRN: <https://ssrn.com/abstract=4810251> or <http://dx.doi.org/10.2139/ssrn.4810251>
- Acharya, Kamal, Online Food Order System (May 2, 2024). Available at
SSRN: <https://ssrn.com/abstract=4814732> or <http://dx.doi.org/10.2139/ssrn.4814732>
- Acharya, Kamal, University management system project. (May 1, 2024). Available at
SSRN: <https://ssrn.com/abstract=4814103> or <http://dx.doi.org/10.2139/ssrn.4814103>
- Acharya, Kamal, Online banking management system. (May 1, 2024). Available at
SSRN: <https://ssrn.com/abstract=4813597> or <http://dx.doi.org/10.2139/ssrn.4813597>
- Acharya, Kamal, Online Job Portal Management System (May 5, 2024). Available at
SSRN: <https://ssrn.com/abstract=4817534> or <http://dx.doi.org/10.2139/ssrn.4817534>
- Acharya, Kamal, Employee leave management system. (May 7, 2024). Available at
SSRN: <https://ssrn.com/abstract=4819626> or <http://dx.doi.org/10.2139/ssrn.4819626>
- Acharya, Kamal, Online electricity billing project report. (May 7, 2024). Available at
SSRN: <https://ssrn.com/abstract=4819630> or <http://dx.doi.org/10.2139/ssrn.4819630>
- Acharya, Kamal, POLICY MANAGEMENT SYSTEM PROJECT REPORT. (December 10, 2023). Available at
SSRN: <https://ssrn.com/abstract=4831694> or <http://dx.doi.org/10.2139/ssrn.4831694>
- Acharya, Kamal, Online job placement system project report. (January 10, 2023). Available at
SSRN: <https://ssrn.com/abstract=4831638> or <http://dx.doi.org/10.2139/ssrn.4831638>
- Acharya, Kamal, Software testing for project report. (May 16, 2023). Available at
SSRN: <https://ssrn.com/abstract=4831028> or <http://dx.doi.org/10.2139/ssrn.4831028>
- Acharya, Kamal, ONLINE CRIME REPORTING SYSTEM PROJECT. (August 10, 2022). Available at

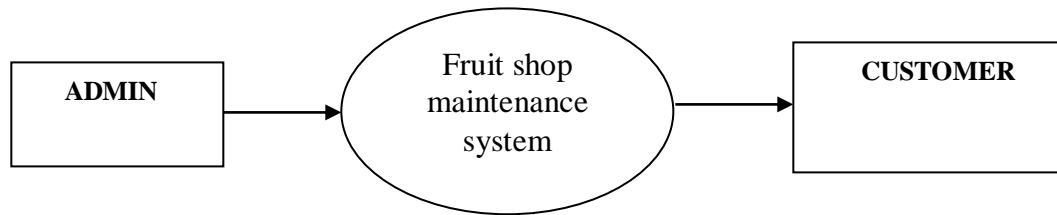
SSRN: <https://ssrn.com/abstract=4831015> or <http://dx.doi.org/10.2139/ssrn.4831015>
 Acharya, Kamal, Barber ordering system project report. (October 10, 2022). Available at
 SSRN: <https://ssrn.com/abstract=4832704> or <http://dx.doi.org/10.2139/ssrn.4832704>
 Acharya, Kamal, Teachers Record Management System Project Report (December 10, 2023). Available at
 SSRN: <https://ssrn.com/abstract=4833821> or <http://dx.doi.org/10.2139/ssrn.4833821>
 Acharya, Kamal, Dairy Management System Project Report (December 20, 2020). Available at
 SSRN: <https://ssrn.com/abstract=4835231> or <http://dx.doi.org/10.2139/ssrn.4835231>

Acharya, Kamal, Electrical Shop Management System Project (December 10, 2019). Available at
 SSRN: <https://ssrn.com/abstract=4835238> or <http://dx.doi.org/10.2139/ssrn.4835238>
 Acharya, Kamal, Online book store management system project report. (February 10, 2020). Available at
 SSRN: <https://ssrn.com/abstract=4835277> or <http://dx.doi.org/10.2139/ssrn.4835277>
 Acharya, Kamal, Paint shop management system project report. (January 10, 2019). Available at
 SSRN: <https://ssrn.com/abstract=4835441> or <http://dx.doi.org/10.2139/ssrn.4835441>
 Acharya, Kamal, Supermarket billing system project report. (August 10, 2021). Available at
 SSRN: <https://ssrn.com/abstract=4835474> or <http://dx.doi.org/10.2139/ssrn.4835474>
 Acharya, Kamal, Online taxi booking system project report. (March 10, 2022). Available at
 SSRN: <https://ssrn.com/abstract=4837729> or <http://dx.doi.org/10.2139/ssrn.4837729>
 Acharya, Kamal, Online car servicing system project report. (March 10, 2023). Available at
 SSRN: <https://ssrn.com/abstract=4837832> or <http://dx.doi.org/10.2139/ssrn.4837832>
 Acharya, Kamal, School management system project report. (July 10, 2021). Available at
 SSRN: <https://ssrn.com/abstract=4837837> or <http://dx.doi.org/10.2139/ssrn.4837837>
 Acharya, Kamal, Furniture Showroom Management System Project Report (March 21, 2021). Available at
 SSRN: <https://ssrn.com/abstract=4839422> or <http://dx.doi.org/10.2139/ssrn.4839422>
 Acharya, Kamal, Online Vehicle Rental System Project Report (March 21, 2019). Available at
 SSRN: <https://ssrn.com/abstract=4839429> or <http://dx.doi.org/10.2139/ssrn.4839429>

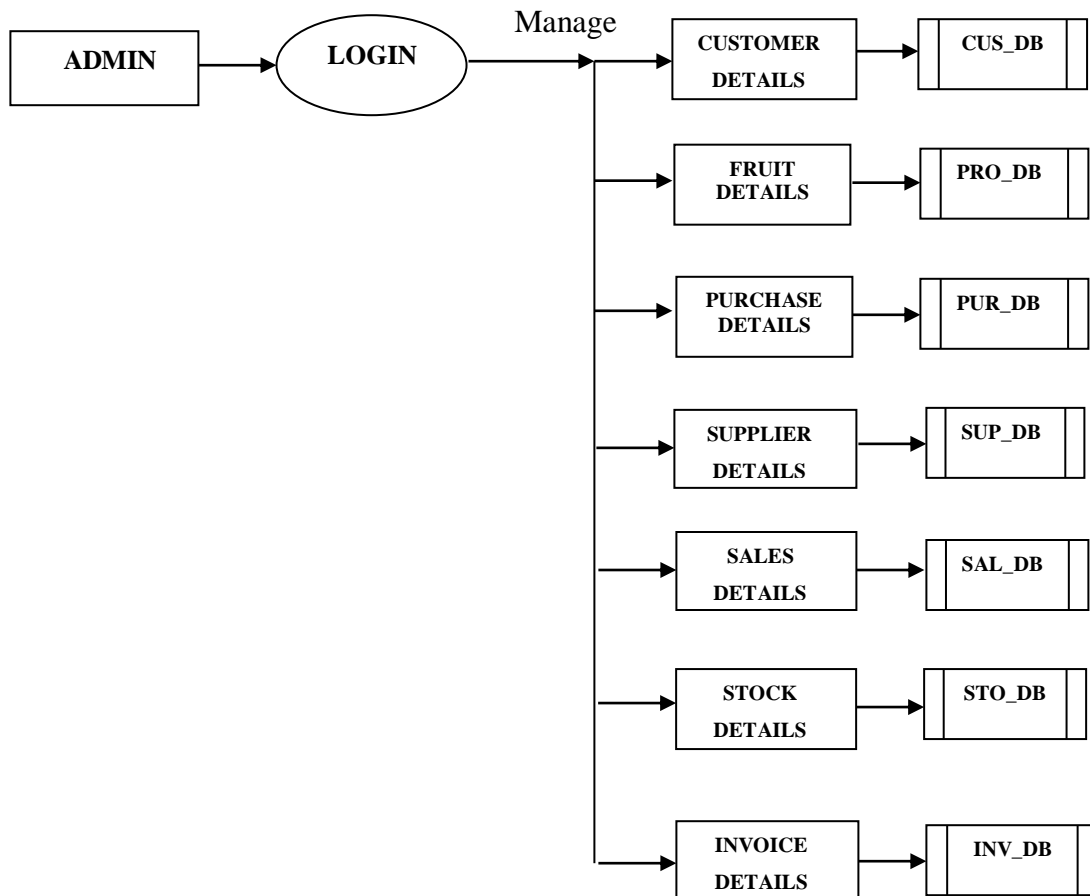
APPENDIX

A. DATA FLOW DIAGRAM

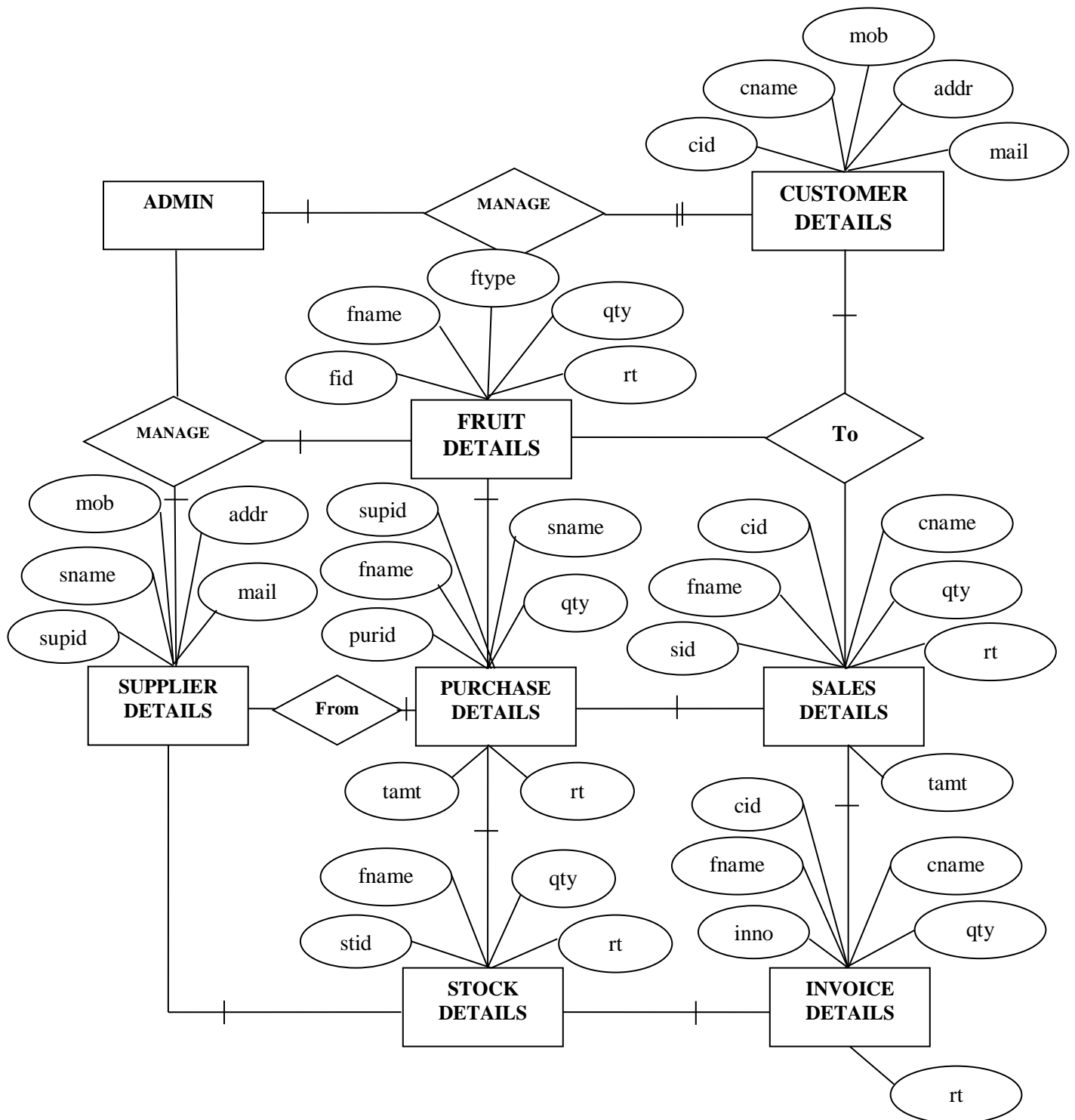
LEVEL 0



LEVEL 1



B. ER DIAGRAM



C. TABLE DESIGN

Table name : customer

Primary key : c_id

Field name	Data type	Width	Description
c_id	int	11	Customer id
cname	varchar	25	Customer name
mob	int	10	Mobile number
addr	char	20	Address
email	varchar	20	Mail id

Table name : Fruit

Primary key : f_id

Field name	Data type	Width	Description
f_id	int	11	Fruit id
fname	varchar	25	Fruit name
ftype	varchar	20	Fruit type
qty	int	10	Quantity
rt	int	10	Rate

Table name : Supplier

Primary key : sup_id

Field name	Data type	Width	Description
sup_id	int	11	Supplier id
sname	varchar	25	Supplier name
mno	int	10	Mobile number
mail	varchar	20	Mail id
addr	char	20	Address

Table name : Purchase

Primary key : pur_id | Foreign Key:f_id, sup_id

Field name	Data type	Width	Description
pur_id	int	11	Purchase id
f_id	int	11	Fruit id
fname	varchar	25	Fruit name
sup_id	int	11	Supplier id
sname	varchar	25	Supplier name
qty	int	10	Quantity
rt	int	10	Rate
tamt	int	10	Total amount

Table name : Sales

Primary key : s_id | Foreign Key: f_id,c_id

Field name	Data type	Description
s_id	int(11)	Sales id
f_id	int(11)	Fruit id
fname	varchar(25)	Fruit name
c_id	int(11)	Customer id
cname	varchar(25)	Customer name
qty	int(10)	Quantity
rt	int(10)	Rate
tamt	int(10)	Total amount

Table name : Stock

Primary key : st_id

Field name	Data type	Width	Description
st_id	int	11	Stock id
f_id	int	11	Fruit id
fname	varchar	25	Fruit name
qty	int	10	Quantity
rt	int	10	Rate

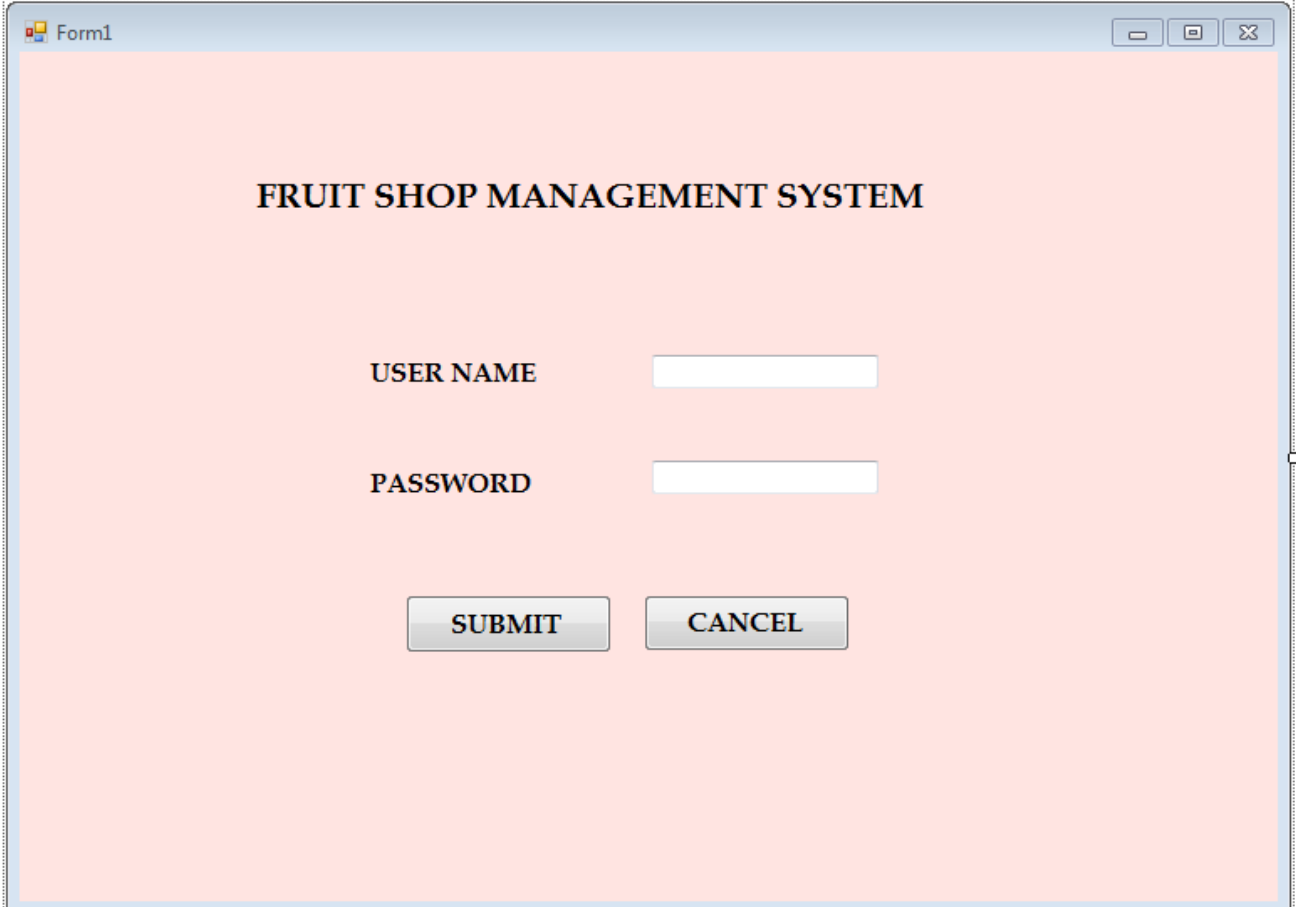
Table name : Billing

Primay key : b_no | Foreign Key: f_id,c_id

Field name	Data type	Width	Description
b_no	int	11	Bill number
f_id	int	11	Fruit id
fname	varchar	25	Fruit name
c_id	int	11	Customer id
cname	varchar	25	Customer name
qty	int	10	Quantity

SCREENSHOT

Admin login



The screenshot shows a web application window titled "Form1". The background is a light pink color. At the top center, the text "FRUIT SHOP MANAGEMENT SYSTEM" is displayed in a bold, black, serif font. Below this, there are two input fields. The first is labeled "USER NAME" and the second is labeled "PASSWORD", both in a bold, black, serif font. Each label is followed by a white rectangular input box. At the bottom of the form, there are two buttons: "SUBMIT" and "CANCEL", both in a bold, black, serif font. The buttons have a light gray gradient and a thin black border. The window has a standard Windows-style title bar with minimize, maximize, and close buttons.

Form1

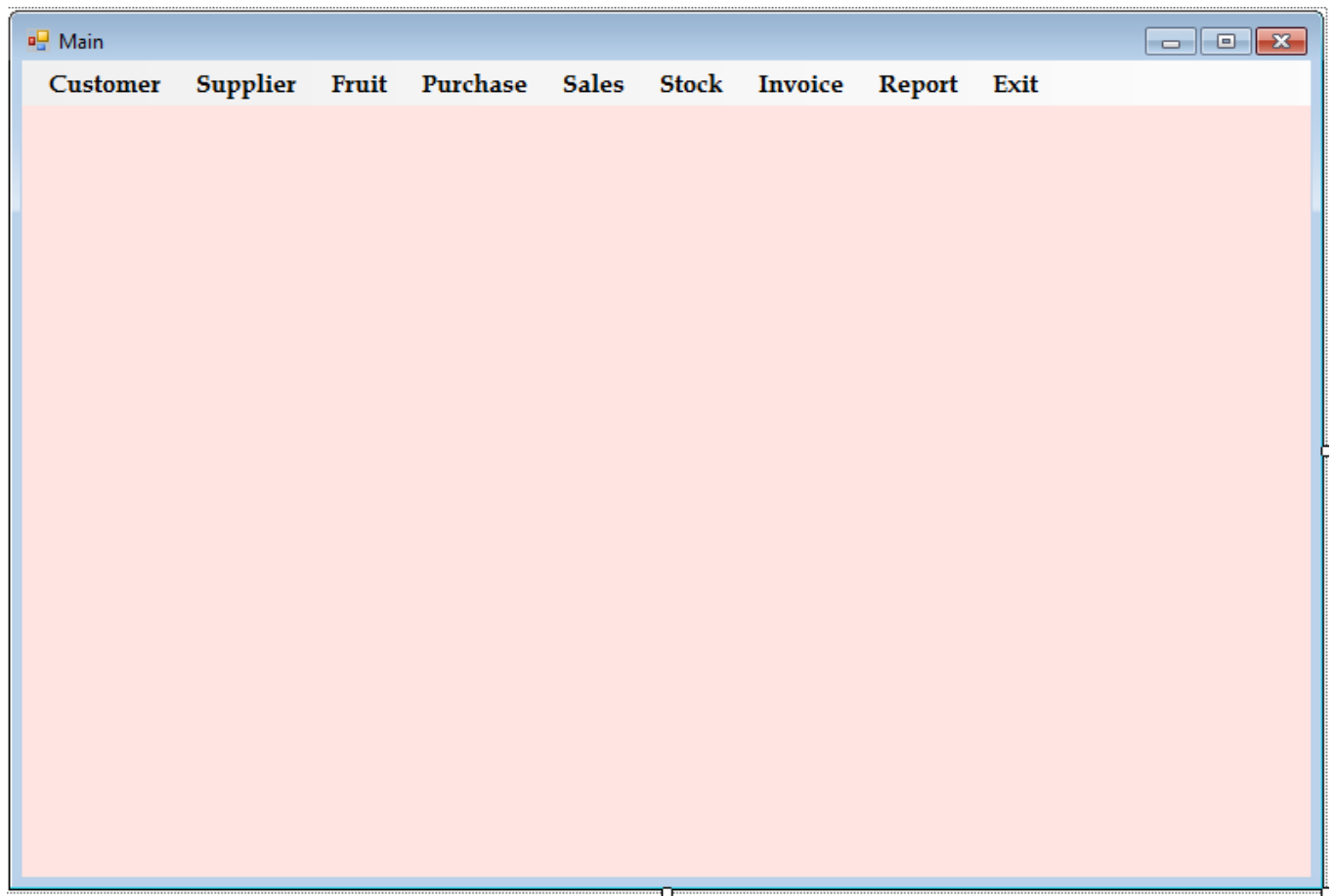
FRUIT SHOP MANAGEMENT SYSTEM

USER NAME

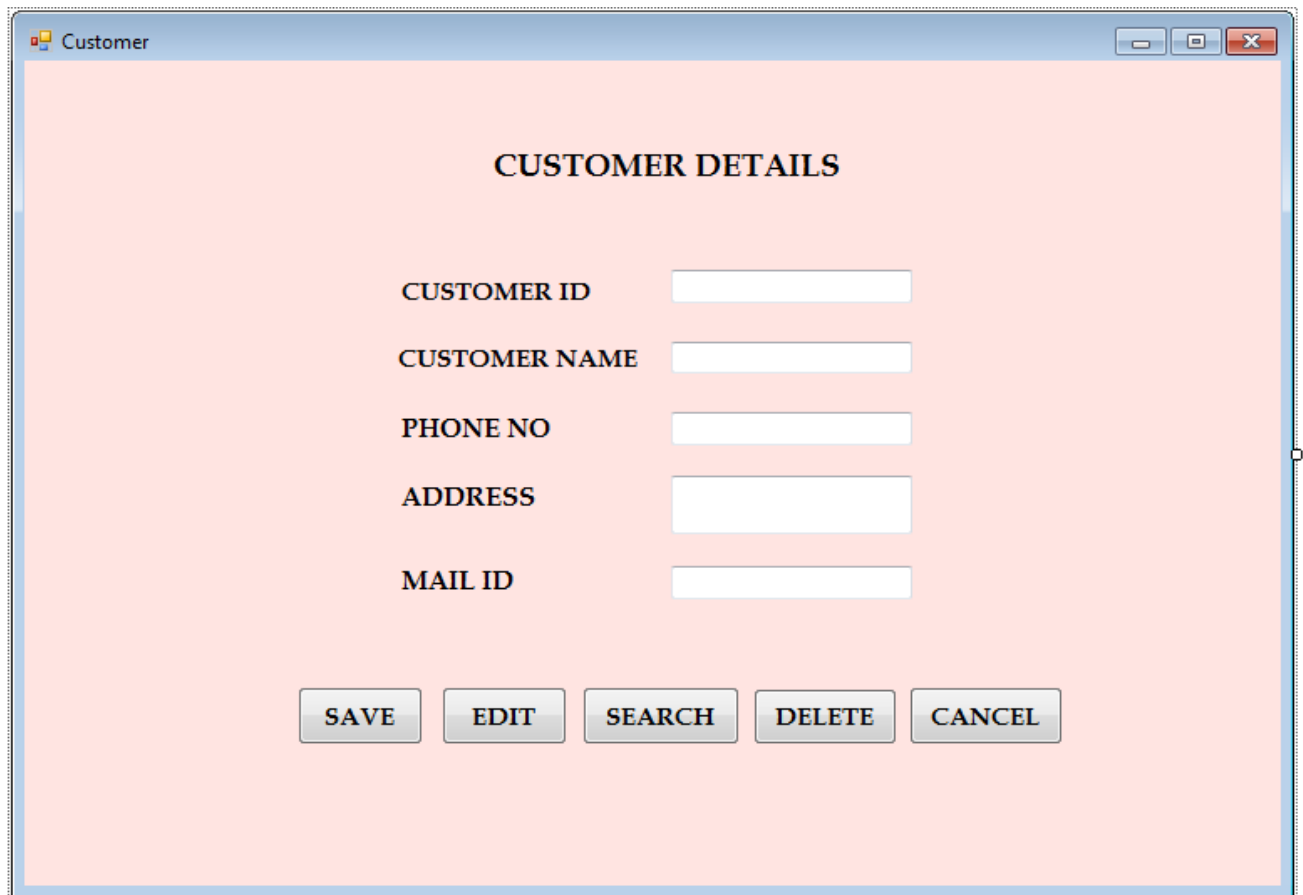
PASSWORD

SUBMIT **CANCEL**

Main Menu



Customer Details



A screenshot of a software window titled "Customer" with a light blue border and standard Windows window controls (minimize, maximize, close). The window has a light pink background. At the top center, the text "CUSTOMER DETAILS" is displayed in a bold, black, serif font. Below this, there are five labels in a bold, black, serif font, each followed by a white rectangular input field: "CUSTOMER ID", "CUSTOMER NAME", "PHONE NO", "ADDRESS", and "MAIL ID". At the bottom of the form, there are five buttons with a grey gradient and black text: "SAVE", "EDIT", "SEARCH", "DELETE", and "CANCEL".

CUSTOMER DETAILS

CUSTOMER ID

CUSTOMER NAME

PHONE NO

ADDRESS

MAIL ID

SAVE **EDIT** **SEARCH** **DELETE** **CANCEL**

Supplier Details



A screenshot of a web application window titled "Supplier". The window has a light blue header bar with standard window controls (minimize, maximize, close) on the right. The main content area has a light pink background. At the top center, the text "SUPPLIER DETAILS" is displayed in a bold, black, serif font. Below this, there are five labels in a bold, black, serif font, each followed by a white rectangular input field: "SUPPLIER ID", "SUPPLIER NAME", "MOBILE NO", "ADDRESS", and "MAIL ID". At the bottom of the form, there are five buttons with a light gray gradient and black text: "SAVE", "EDIT", "SEARCH", "DELETE", and "CANCEL".

SUPPLIER DETAILS

SUPPLIER ID

SUPPLIER NAME

MOBILE NO

ADDRESS

MAIL ID

SAVE **EDIT** **SEARCH** **DELETE** **CANCEL**

Fruit Details

Product

FRUIT DETAILS

FRUIT ID

FRUIT NAME

FRUIT TYPE

QUANTITY

AMOUNT

SAVE

EDIT

SEARCH

DELETE

CANCEL

Purchase Details

Purchase

PURCHASE DETAILS

PURCHASE ID

SUPPLIER NAME

FRUIT ID

QUANTITY

FRUIT NAME

PRICE

FRUIT TYPE

TOTAL AMOUNT

SUPPLIER ID

SAVE

EDIT

SEARCH

DELETE

CANCEL

Sales Details

Sales

SALES DETAILS

SALES ID

FRUIT TYPE

CUSTOMER ID

QUANTITY

CUSTOMER NAME

PRICE

FRUIT ID

AMOUNT

FRUIT NAME

SAVE

EDIT

SEARCH

DELETE

CANCEL

Stock Details

A screenshot of a software window titled "Stock". The window has a light blue title bar with standard minimize, maximize, and close buttons. The main content area has a light pink background. At the top center, the text "STOCK DETAILS" is displayed in a bold, black, serif font. Below this, there are five labels on the left, each followed by a form field on the right: "STOCK ID" with a text input field, "PRODUCT NAME" with a dropdown menu, "PRODUCT TYPE" with a dropdown menu, "QUANTITY" with a text input field, and "AMOUNT" with a text input field. At the bottom of the window, there are five buttons arranged horizontally: "SAVE", "EDIT", "SEARCH", "DELETE", and "CANCEL". All buttons have a light gray gradient and a thin black border.

STOCK DETAILS

STOCK ID

PRODUCT NAME

PRODUCT TYPE

QUANTITY

AMOUNT

SAVE **EDIT** **SEARCH** **DELETE** **CANCEL**

SAMPLE CODE

```
Imports System.Data
Imports System.Data.SqlClient
Public Class Customer
    Dim con As New SqlConnection("Data Source=SYSTEM-6\SQLEXPRESS;Initial
Catalog=icecream;Persist Security Info=True;User ID=sa;Password=sql")
    Dim qry As String
    Dim com As New SqlCommand
    Dim adp As New SqlDataAdapter
    Dim ds As New DataSet
    Dim i As Integer
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or TextBox4.Text =
"" Or TextBox5.Text = "" Then
            MsgBox("Please Fill All Details", MsgBoxStyle.Information)
        Else

            qry = "insert into customer values(" + TextBox1.Text + "," + TextBox2.Text + "," +
TextBox3.Text + "," + TextBox4.Text + "," + TextBox5.Text + ")"
            com = New SqlCommand(qry, con)
            con.Open()
            com.ExecuteNonQuery()
            con.Close()
            MsgBox("Successfully Saved", MsgBoxStyle.Information)
        End If

    Try
        Dim p As String
        qry = "select cid from customer"
        com = New SqlCommand(qry, con)
```

```

adp = New SqlDataAdapter(com)
ds = New DataSet
adp.Fill(ds, "customer")
i = ds.Tables("customer").Rows.Count
p = ds.Tables("customer").Rows(i - 1)(0)
p = Mid(p, 2)
p = Val(p) + 1
If Len(p) = 1 Then
    p = "00" + p
ElseIf Len(p) = 2 Then
    p = "0" + p
End If
TextBox1.Text = "C" + p

```

```

Catch ex As Exception

```

```

    TextBox1.Text = "C001"

```

```

End Try

```

```

TextBox2.Text = ""

```

```

TextBox3.Text = ""

```

```

TextBox4.Text = ""

```

```

TextBox5.Text = ""

```

```

End Sub

```

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

```

```

Handles Button2.Click

```

```

    If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or TextBox4.Text = "" Or TextBox5.Text = "" Then

```

```

        MsgBox("Please Search Any One Details Then Edit", MsgBoxStyle.Information)

```

```

    Else

```

```

        qry = "update customer set cname=" + TextBox2.Text + ",pno=" + TextBox3.Text + ",addr=" + TextBox4.Text + ",mail=" + TextBox5.Text + "where cid=" + TextBox1.Text + ""

```

```

com = New SqlCommand(qry, con)
con.Open()
com.ExecuteScalar()
con.Close()
MsgBox("Successfully Updated", MsgBoxStyle.Information)
End If

```

```

TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
TextBox5.Text = ""

```

Try

```

Dim p As String
qry = "select cid from customer"
com = New SqlCommand(qry, con)
adp = New SqlDataAdapter(com)
ds = New DataSet
adp.Fill(ds, "customer")
i = ds.Tables("customer").Rows.Count
p = ds.Tables("customer").Rows(i - 1)(0)
p = Mid(p, 2)
p = Val(p) + 1
If Len(p) = 1 Then
    p = "00" + p
ElseIf Len(p) = 2 Then
    p = "0" + p
End If
TextBox1.Text = "C" + p

```

Catch ex As Exception

```

TextBox1.Text = "C001"

```

End Try

Try

```
qry = "select * from customer"  
com = New SqlCommand(qry, con)  
adp = New SqlDataAdapter(com)  
ds = New DataSet  
adp.Fill(ds, "customer")
```

Catch ex As Exception

```
MsgBox(ex.Message)
```

End Try

End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Handles Button3.Click

Try

```
Dim s As String  
s = InputBox("Enter Customer id")  
qry = "select * from customer where cid='" + s + "'"<br>com = New SqlCommand(qry, con)  
adp = New SqlDataAdapter(com)  
ds = New DataSet  
adp.Fill(ds, "customer")  
TextBox1.Text = ds.Tables("customer").Rows(0)(0)  
TextBox2.Text = ds.Tables("customer").Rows(0)(1)  
TextBox3.Text = ds.Tables("customer").Rows(0)(2)  
TextBox4.Text = ds.Tables("customer").Rows(0)(3)  
TextBox5.Text = ds.Tables("customer").Rows(0)(4)
```

Catch ex As Exception

```
MsgBox(ex.Message)
```

```
End Try
```

```
End Sub
```

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles Button4.Click
```

```
con.Open()
```

```
qry = "delete from customer where cid=" & TextBox1.Text & " "
```

```
com = New SqlCommand(qry, con)
```

```
com.ExecuteNonQuery()
```

```
MsgBox("successfully deleted")
```

```
con.Close()
```

```
TextBox1.Text = ""
```

```
TextBox2.Text = ""
```

```
TextBox3.Text = ""
```

```
TextBox4.Text = ""
```

```
TextBox5.Text = ""
```

```
End Sub
```

```
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles Button5.Click
```

```
Me.Hide()
```

```
main.Show()
```

```
End Sub
```

```
Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles TextBox1.TextChanged
```

```
End Sub
```

```
Private Sub Customer_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles MyBase.Load
```

```

End Sub

End Class

Imports System.Data
Imports System.Data.SqlClient
Public Class Purchase
    Dim con As New SqlConnection("Data Source=SYSTEM-6\SQLEXPRESS;Initial
Catalog=icecream;Persist Security Info=True;User ID=sa;Password=sql")
    Dim qry, qry1 As String
    Dim com As New SqlCommand
    Dim adp, adp1 As New SqlDataAdapter
    Dim ds As New DataSet
    Dim i As Integer
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        If TextBox1.Text = "" Or ComboBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text
= "" Or ComboBox2.Text = "" Or TextBox4.Text = "" Or TextBox5.Text = "" Or TextBox6.Text
= "" Or TextBox7.Text = "" Then
            MsgBox("Please Fill All Details", MsgBoxStyle.Information)
        Else

            qry = "insert into purchase values(" + TextBox1.Text + "," + ComboBox1.Text + "," +
TextBox2.Text + "," + TextBox3.Text + "," + ComboBox2.Text + "," + TextBox4.Text + "," +
+ TextBox5.Text + "," + TextBox6.Text + "," + TextBox7.Text + ")"
            com = New SqlCommand(qry, con)
            con.Open()
            com.ExecuteScalar()
            con.Close()
            MsgBox("Successfully Saved", MsgBoxStyle.Information)
        End If

    Try
        Dim p As String
        qry = "select purid from purchase"
        com = New SqlCommand(qry, con)
        adp = New SqlDataAdapter(com)
        ds = New DataSet
        adp.Fill(ds, "purchase")
        i = ds.Tables("purchase").Rows.Count
        p = ds.Tables("purchase").Rows(i - 1)(0)
        p = Mid(p, 2)
        p = Val(p) + 1
        If Len(p) = 1 Then
            p = "00" + p
        ElseIf Len(p) = 2 Then
            p = "0" + p
        End If
    End Try
End Class

```

```

End If
TextBox1.Text = "P" + p

Catch ex As Exception
    TextBox1.Text = "P001"
End Try
ComboBox1.Text = ""
ComboBox2.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
TextBox5.Text = ""
TextBox6.Text = ""
TextBox7.Text = ""
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
    If TextBox1.Text = "" Or ComboBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text
= "" Or ComboBox2.Text = "" Or TextBox4.Text = "" Or TextBox5.Text = "" Or TextBox6.Text
= "" Or TextBox7.Text = "" Then
        MsgBox("Please Search Any One Details Then Edit", MsgBoxStyle.Information)
    Else
        qry = "update purchase set pid=" + ComboBox1.Text + ",pname=" + TextBox2.Text +
        ",ptype=" + TextBox3.Text + ",supid=" + ComboBox2.Text + ",supname=" + TextBox4.Text
        + ",qty=" + TextBox5.Text + ",price=" + TextBox6.Text + ",tamt=" + TextBox7.Text +
        "where purid=" + TextBox1.Text + ""
        com = New SqlCommand(qry, con)
        con.Open()
        com.ExecuteScalar()
        con.Close()
        MsgBox("Successfully Updated", MsgBoxStyle.Information)
    End If
    ComboBox1.Text = ""
    TextBox2.Text = ""
    TextBox3.Text = ""
    ComboBox2.Text = ""
    TextBox4.Text = ""
    TextBox5.Text = ""
    TextBox6.Text = ""
    TextBox7.Text = ""
    Try
        Dim p As String
        qry = "select purid from purchase"
        com = New SqlCommand(qry, con)
        adp = New SqlDataAdapter(com)

```



```

ds = New DataSet
adp.Fill(ds, "purchase")
i = ds.Tables("purchase").Rows.Count
p = ds.Tables("purchase").Rows(i - 1)(0)
p = Mid(p, 2)
p = Val(p) + 1
If Len(p) = 1 Then
    p = "00" + p
ElseIf Len(p) = 2 Then
    p = "0" + p
End If
TextBox1.Text = "P" + p

```

```

Catch ex As Exception
    TextBox1.Text = "P001"
End Try

```

```

Try
    qry = "select * from purchase"
    com = New SqlCommand(qry, con)
    adp = New SqlDataAdapter(com)
    ds = New DataSet
    adp.Fill(ds, "purchase")

```

```

Catch ex As Exception
    MsgBox(ex.Message)
End Try
End Sub

```

```

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click

```

```

Try
    Dim s As String
    s = InputBox("Enter purchase id")
    qry = "select * from purchase where purid=" + s + ""
    com = New SqlCommand(qry, con)
    adp = New SqlDataAdapter(com)
    ds = New DataSet
    adp.Fill(ds, "purchase")
    TextBox1.Text = ds.Tables("purchase").Rows(0)(0)
    ComboBox1.Text = ds.Tables("purchase").Rows(0)(1)
    TextBox2.Text = ds.Tables("purchase").Rows(0)(2)
    TextBox3.Text = ds.Tables("purchase").Rows(0)(3)
    ComboBox2.Text = ds.Tables("purchase").Rows(0)(4)
    TextBox4.Text = ds.Tables("purchase").Rows(0)(5)
    TextBox5.Text = ds.Tables("purchase").Rows(0)(6)

```

```

        TextBox6.Text = ds.Tables("purchase").Rows(0)(7)
        TextBox7.Text = ds.Tables("purchase").Rows(0)(8)
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub

```

```

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
    con.Open()
    qry = "delete from purchase where purid=" & TextBox1.Text & " "
    com = New SqlCommand(qry, con)
    com.ExecuteNonQuery()
    MsgBox("successfully deleted")
    con.Close()
    TextBox1.Text = ""
    ComboBox1.Text = ""
    TextBox2.Text = ""
    TextBox3.Text = ""
    ComboBox2.Text = ""
    TextBox4.Text = ""
    TextBox5.Text = ""
    TextBox6.Text = ""
    TextBox7.Text = ""
End Sub

```

```

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
    Me.Hide()
    main.Show()
End Sub

```

```

Private Sub Purchase_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    fill()
    fresh()
End Sub
Private Sub fill()
    Try
        qry = "select * from product"
        con.Open()
        com = New SqlCommand(qry, con)
        con.Close()
        adp = New SqlDataAdapter(com)
        ds = New DataSet()
        adp.Fill(ds, "product")
    End Try
End Sub

```

```

        i = ds.Tables("product").Rows.Count
        For j = 0 To i - 1
            ComboBox1.Items.Remove(ds.Tables("product").Rows(j)(0))
            ComboBox1.Items.Add(ds.Tables("product").Rows(j)(0))
        Next
    Catch ex As Exception
    End Try
End Sub
Private Sub fresh()
    Try

        qry1 = "select * from supplier"
        con.Open()
        com = New SqlCommand(qry1, con)
        con.Close()
        adp1 = New SqlDataAdapter(com)
        ds = New DataSet()
        adp1.Fill(ds, "supplier")
        i = ds.Tables("supplier").Rows.Count
        For j = 0 To i - 1
            ComboBox2.Items.Remove(ds.Tables("supplier").Rows(j)(0))
            ComboBox2.Items.Add(ds.Tables("supplier").Rows(j)(0))
        Next
    Catch ex As Exception

    End Try
End Sub

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ComboBox1.SelectedIndexChanged
    Try
        qry = "select * from product where pid='" + ComboBox1.Text.ToString() + "'"
        com = New SqlCommand(qry, con)
        con.Open()
        adp = New SqlDataAdapter(com)
        con.Close()
        ds = New DataSet()
        adp.Fill(ds, "product")
        TextBox2.Text = ds.Tables("product").Rows(0)(1)
        TextBox3.Text = ds.Tables("product").Rows(0)(2)
    Catch ex As Exception
    End Try
End Sub

Private Sub ComboBox2_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ComboBox2.SelectedIndexChanged

```

```

Try
    qry1 = "select * from supplier where supid=" + ComboBox2.Text.ToString() + ""
    com = New SqlCommand(qry1, con)
    con.Open()
    adp1 = New SqlDataAdapter(com)
    con.Close()
    ds = New DataSet()
    adp1.Fill(ds, "supplier")
    TextBox4.Text = ds.Tables("supplier").Rows(0)(1)
Catch ex As Exception
End Try
End Sub

Private Sub TextBox7_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox7.TextChanged
    TextBox7.Text = Val(TextBox5.Text) * Val(TextBox6.Text)
End Sub
End Class

```